VOMS & Reliability

Vincenzo Ciaschini & Andrea Ceccanti

WLCG Service Reliability Workshop, 20-30/11/07

Presenter: Francesco Giacomini

www.eu-egee.org

Information Society and Media

Reliability:

- **A reliable service requires two things:**
  - Reliability by Design
  - Reliability by Coding

VOMS – Reliable by Design

- **A Stateless service:**
  - No information is retained between connections
  - Allows connections to be done to different servers
- **Replication-friendly**
  - See above
  - CLI clients have built-in support for replicas
- **Maximum separation among queries**
  - Use processes rather than threads
  - Problems with one execution stay contained
- **Stay away from non-standard features**
  - Ensures consistent behaviour
- **Keep the data in a DB**
  - Takes advantage of DB reliability

VOMS – Reliable by coding

- **Always check function return codes**
  - Should be obvious, but…
- **Always check function parameters for validity**
  - All functions!
  - Slight performance hit…
  - But is protection against internal errors.
- **Reduce complexity**
  - Use C++ features for memory allocation
- **Free resources as soon as possible**
- **Stay away from function side effects as much as possible**
  - One way in, one way out
  - Implies: stay away from exceptions
    - Exception safety becomes a non-issue
    - Portability problems with exceptions.
    - Except for constructors, obviously!

VOMS-Admin

- **J2EE Web application**
  - leverage J2EE container for replication / load balancing
- **Designed for robustness / stability / performance**
  - Leverage Model/View/Controller design pattern
    - clear separation between data, presentation and control
  - Always validate user input
    - do not trust anything coming from the user
  - Leverage succesful existing frameworks
    - to manage data (Hibernate)
    - to manage presentation and flow control (Apache Struts)